#18

PATENT
Attorney Docket No. COS97033

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

John V. MCLAIN Jr. et al.

Serial No.: 08/987,849

Filed: December 9, 1997

For: SYSTEM AND METHOD FOR MANAGING COMPUTER SYSTEM RESOURCES USING COMMAND CONTROL VECTORS

Group Art Unit: 2126

Examiner: S. Lao



## APPEAL BRIEF

U.S. Patent and Trademark Office
2011 South Clark Place
Customer Window, Mail Stop Appeal Brief – Patents
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

Sir:

This Appeal Brief is submitted in response to the rejection mailed September 24, 2003 and in support of the Notice of Appeal filed December 23, 2003.

## I. REAL PARTY IN INTEREST

The real party in interest in this appeal is WorldCom, Inc.

## II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals and interferences.

## III. STATUS OF CLAIMS

Claims 1-6 and 14-18 are pending in this application. All of the pending claims are the subject of the present appeal.

## IV. STATUS OF AMENDMENTS

No Amendment has been filed subsequent to the Office Action mailed September 24, 2003. Appellants, however, note that in the Office Action, claims 1-6 and 14-18 were rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-23 of U.S. Patent No. 6,295,518 and claims 1-18 of U.S. Patent No. 6,256,659. While not concurring with the rejections, but to simplify the issues for Appeal, Appellants have filed Terminal Disclaimers concurrently herewith this Appeal Brief to obviate the obviousness-type double patenting rejections based on U.S. Patents Nos. 6,295,518 and 6,256,659. Accordingly, Appellants respectfully request withdrawal of the obviousness-type double patenting rejections.

## V. SUMMARY OF THE INVENTION

The present invention is directed to managing computer system resources (specification – page 1, lines 6-8). As described in Appellants' specification at page 4, lines 1-25 and page 14, lines 1-20 and with reference to Fig. 2, managing the computer system resources includes processing multiple tasks using command control vectors (CCVs) and a single copy of a

method object. Command response manager 216 may employ CCVs to control processing of input messages and may allocate a CCV for each input command to be processed. Each CCV may maintain a variety of data, including status information and pointers that identify memory objects, such as command response tables and scripts and pointers that identify data objects that store data associated with the execution of a method object.

As further described at page 19, line 1 to page 20, line 7, a CCV identifies a task or thread that requires processing time and includes pointers to method objects and data objects that are to be used for processing the task. Method objects may include command response tables and scripts. Referring to Fig. 9, an exemplary CCV 910 may include fields 912-924. Field 912 may identify a source of external communication, field 914 may provide additional details associated with field 912, field 916 may contain a pointer to a method object that contains instructions for processing the task and field 918 may contain a pointer to a particular instruction within the method object or command response table identified in field 916. Field 920 points to a script, field 922 provides a pointer to a data segment of a data object and field 924 stores optional pointers and flags.

The CCVs may permit more than one thread or task to point to the same copy of a command response table, script or other method object (specification - page 20, lines 8-16). Therefore, CCVs permit multiple tasks to be processed using a single copy of a command response table, script or other method object. Using CCVs in this manner saves considerable memory space and time as compared to systems in which duplicate copies of the method object must be stored and retrieved. In addition, by providing pointer field 918, which maintains a

pointer to a current instruction, if processing of a CCV is interrupted and processing is later resumed, processing can resume exactly where it was interrupted (specification - page 21, lines 1-12).

## VI. ISSUES

Whether claims 1-3 and 14-17 are unpatentable under 35 U.S.C. § 103(a) over Burgess (U.S. Patent No. 5,652,888) in view of "Design Patterns, Elements of Reusable Object-Oriented Software," pages 127-134 by Erich Gamma et al. (hereinafter Gamma);

Whether claims 4 and 18 are unpatentable under 35 U.S.C. § 103(a) over Burgess in view of Gamma and further in view of admitted prior art; and

Whether claims 5 and 6 are unpatentable under 35 U.S.C. § 103(a) over Burgess in view of Gamma, the admitted prior art and further in view of "Compiling Distributed C++," by H. Carr et al. (hereinafter Carr).

## VII. GROUPING OF CLAIMS

Appellants are satisfied to let claims 1 and 14 stand or fall together. Appellants are also satisfied to let claims 2 and 16 stand or fall together, claims 3 and 17 stand or fall together and claims 4 and 18 stand or fall together. Each of claims 5, 6 and 15 do not stand or fall together with any of the other claims for the reasons discussed in the Argument section below.

VIII. ARGUMENT

A. The Rejections

1. Claims 1-3 and 14-17 are patentable over Burgess in view of Gamma.

The initial burden of establishing a *prima facie* basis to deny patentability to a claimed invention always rests upon the Examiner. In re Oetiker, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In rejecting a claim under 35 U.S.C. § 103, the Examiner must provide a factual basis to support the conclusion of obviousness. In re Warner, 379 F.2d 1011, 154 USPQ 173 (CCPA 1967). Based upon the objective evidence of record, the Examiner is required to make the factual inquiries mandated by Graham v. John Deere Co., 86 S.Ct. 684, 383 U.S. 1, 148 USPQ 459 (1966). The Examiner is also required to explain how and why one having ordinary skill in the art would have been realistically motivated to modify an applied reference and/or combine applied references to arrive at the claimed invention. Uniroyal, Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 5 USPQ2d 1434 (Fed. Cir. 1988).

In establishing the requisite motivation, it has been consistently held that the requisite motivation to support the conclusion of obviousness is not an abstract concept, but must stem from the prior art as a whole to impel one having ordinary skill in the art to modify a reference or to combine references with a reasonable expectation of successfully achieving some particular realistic objective. See, for example, Interconnect Planning Corp. v. Feil, 227 USPQ 543 (Fed. Cir. 1985). Consistent legal precedent admonishes against the indiscriminate combination of prior art references. Carella v. Starlight Archery, 804 F.2d 135, 231 USPQ 644 (Fed. Cir. 1986); Ashland Oil, Inc. v. Delta Resins & Refractories, Inc., 776 F.2d 281, 227 USPQ 657 (Fed. Cir. 1985).

a) Claims 1 and 14

With these principles in mind, the arguments below use claim 1 as representative of the group of claims including claims 1 and 14. Claim 1 recites a method for managing computer system resources that includes generating a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message. Claim 1 also recites that the generating a first command control vector comprises identifying the method object in the first command control vector and identifying, in the first command control vector, a first current instruction of the method object, wherein the first current instruction is used to process the first input message.

The Office Action states that Burgess discloses these features and points to col. 4, lines 3-19 and col. 7, line 55 to col. 9, line 21 for support (Office Action – page 5). Appellants respectfully disagree.

First, Burgess is directed to a method and system for interconnecting visual software components and passing messages between connected components. Visual programs allow various components, such as a scroll bar, to be interconnected visually to effect program behavior (Burgess – col. 1, lines 15-50). Burgess at col. 4, lines 3-19 discloses that a source object sends a message/communicates the occurrence of an event by invoking a member function of a target object. This portion of Burgess further discloses that each input and output port is preferably associated with a function member of the run-time object. For example, a scroll bar object may have a function member ValueIn corresponding to an input port for receiving a new value and a function member ValueOut corresponding to an output port for sending a new value. The member function ValueOut invokes the target member function corresponding to each input port

to which it is connected and the target member function processes the value information which is passed as a parameter.

This portion of Burgess does not disclose generating a first command control vector for a first input message, as recited in claim 1. Rather, this portion of Burgess merely discloses that when a change occurs, such as a change in the value for a scroll bar when a slider on the scroll bar is moved (see Fig. 1 of Burgess), a member function invokes a target member function to process the value information associated with the change. Invoking a target member function to process a change is not equivalent to generating a first command control vector for a first input message, where the first command control vector identifies a method object that contains or more instructions for processing the first input message.

Burgess at col. 7, line 55 to col. 9, line 21 discloses that the use of an event object allows a connection manager to invoke input member functions with arbitrary parameters (col. 7, line 55 to col. 8, line 4). This portion of Burgess provides pseudocode that may be used to describe the event classes (col. 8, lines 5-45) and pseudocode for a sample scroll bar object (col. 8, line 46 to col. 10, line 60). These portions of Burgess do not disclose or suggest generating a first command control vector for a first input message, much less that the generating includes identifying the method object in the first command control vector and identifying a first current instruction of the method object, where the first current instruction is used to process the first input message, as required by claim 1.

More particularly, the Office Action indicates that Burgess discloses "identifying the method object (ptarget) in the first command control vector (object message format, col. 8, lines 5-15), identifying in the first command control vector (col. 8, line 15), a first current instruction

(pmftarget) used to process the first input message (class CIntEvent, col. 8, lines 13-19)" (Office Action – page 5). These portions of Burgess do not read on or suggest the claimed features.

For example, Burgess at col. 8, lines 5-19 discloses:

> Code Table 2 contains pseudocode that describes the event classes. Each event class derives from the class CEvent. The event class CIntEvent is used to pass information stored as an integer to a target member function. The member function Dispatch is passed a pointer the target object and a pointer to the member function. The member function Dispatch invokes the target member function by the following statement:
>
> (ptarget →*(LongToMember(pmftarget)))(numValue)
>
> The event class CActionEvent is used to pass messages with no corresponding data, and the event class CStringEvent is used to pass messages with string data.

The portion of Burgess, as best understood by Appellants, discloses that "pmftarget," is a pointer to the target member function (see Burgess at col. 7, lines 12-20). Such a pointer to a target member function does not identify a first current instruction of a method object, as recited in claim 1. Rather, this pointer merely points to a particular function. Therefore, this portion of Burgess does not disclose or suggest generating a first command control vector, where the generating comprises identifying a first current instruction of the method object, as recited in claim 1.

Claim 1 also recites generating a second command control vector associated with a second command input message, where the generating a second command control vector includes identifying the same method object identified by the first command control vector in the second command control vector and identifying, in the second command control vector, a second current instruction of the method object, wherein the second current instruction is used to process the second input message. The Office Action states that Burgess discloses these features and

points to col. 4, lines 3-19 and col. 8, line 46 to col. 11, line 12 for support (Office Action – page 5). Appellants respectfully disagree.

Burgess at col. 4, lines 3-19, as discussed above, merely discloses invoking a target member function to process changes, such as changes in the value for a scroll bar when a slider on the scroll bar is moved. Invoking a target member function to process a change is not equivalent to generating a second command control vector associated with a second input message, where the second command control vector identifies the same method object identified by the first command control vector.

Apparently, the Examiner considers the target member function "ptarget" to be equivalent to the claimed same method object identified by the first command control vector and the pointer to the member function "pmftarget" to identify a second current instruction (Office Action – page 5). Appellant respectfully disagrees.

Burgess at col. 8, lines 5-15 discloses that the member function "Dispatch" invokes the target member function by the following statement:

(ptarget →*(LongToMember(pmftarget)))(numValue)

The target member function "ptarget," as best understood by Appellants, is merely used to invoke a particular target member function. That is, "ptarget" points to the target member function. Ptarget, however, does not identify a same method object identified by a first command control vector, as required by claim 1. Further, the term "pmftarget," as best understood by Appellants, is a pointer to a member function target. The pointer to the member function target does not identify a second current instruction of the method object, where the second current instruction is used to process the second input message, as required by claim 1.

The other portions of Burgess relied upon as allegedly disclosing these features, (i.e., col. 8, line 46 to col. 11, line 12) also do not disclose or suggest these features. For example, Burgess at col. 8, line 46 to col. 11, line 12 refers to Code Table 3 that contains pseudocode for a sample scrollbar object. This portion of Burgess does not disclose or suggest generating a second command control vector that identifies the same method object identified by the first command control vector or that the generating a second command control vector includes identifying a second current instruction of the method object, wherein the second current instruction is used to process the second input message, as recited in claim 1.

Even if, for the sake of argument, Burgess could reasonably be construed to disclose generating first and second command control vectors for first and second input messages that identify a same method object, Burgess clearly does not disclose or suggest identifying a first and second current instruction in the first and second command control vectors, respectively, where the first instruction is used to process the first message and the second instruction is used to process the second input message, as required by claim 1.

Gamma was used in the rejection of claim 1 as allegedly disclosing providing a single copy of a class object (Office Action – page 6). Gamma may disclose that a class in an object creation environment may have only one instance. Gamma, however, does not make up for the deficiencies in Burgess with respect to claim 1 discussed above. That is, Gamma does not disclose generating first and second command control vectors that identify the same method object or identifying first and second current instructions of the method object, where the first and second current instructions are used to process the first and second messages, respectively, as recited in claim 1.

Therefore, as a factual matter, the combination of Burgess and Gamma does not disclose or suggest each of the features of claim 1.

In addition, even if, for the sake of argument, the combination of Burgess and Gamma could reasonably be construed to disclose each of the features of claim 1, the Office Action does not provide the motivation required under 35 U.S.C. § 103 as to why it would have been obvious to combine these references.

For example, the Office Action states that Burgess "desires to hide different implementations/functional prototypes of the target object from the source object such that each source member function invokes target member functions with the same prototype" and points to col. 4, lines 31-37 for support (Office Action – page 6). The Office Action further states that Burgess does not provide a mechanism for calling with the same prototype, but that Gamma discloses a mechanism for calling a target object with the same prototype. The Office Action then states that one of ordinary skill in the art would have been motivated to "use the mechanism of Gamma to achieve the hiding in Burgess" (Office Action – page 6). Appellants respectfully disagree.

First, Burgess at col. 4, lines 31-37 discloses that different implementations of event objects are used to represent different function prototypes of target member functions. Each class of event object contains information that is passed to a target member function according to the function prototype and therefore, each source member function invokes target member functions with the same prototype. Therefore, the allegation that Burgess "desires to hide different implementations/functional prototypes of the target object from the source object" is not supported by Burgess at col. 4, lines 31-37.

In addition, even if Burgess did disclose some desire to hide different implementations/functional prototypes, as alleged, the Office Action has still not provided any objective motivation as to why it would have been obvious to combine Gamma with Burgess. The Office Action merely indicates that Gamma provides a mechanism for calling a target object with the same prototype and therefore alleges that one of ordinary skill in the art would have been motivated to use the mechanism of Gamma to achieve the hiding in Burgess. The mere fact that one reference allegedly provides some missing disclosure with respect to a claim does not satisfy the requirements of 35 U.S.C. § 103 as to why it would have been obvious to combine these references.

Appellants assert that it would not have been obvious to combine these two references without the benefit of Appellants' disclosure. For example, the mere fact that both references involve object oriented programming does not establish the requisite realistic motivation to combine the references to arrive at the specifically claimed invention.

Further, the reasoning provided for combining Burgess and Gamma (i.e., to achieve hiding) is merely a conclusory statement regarding an alleged benefit of the combination. It is apparent that the Examiner's approach to the ultimate legal conclusion of obviousness under 35 U.S.C.§ 103 amounts to a retrospective assessment as to how the claimed invention works and then combining unrelated references to arrive at the claimed invention. This type of reverse engineering approach to the obviousness issue under 35 U.S.C. § 103 has been repeatedly judicially condemned. Uniroyal, Inc. v. Rudkin-Wiley Corp., supra; Panduit Corp. v. Dennison Mfg. Co., supra. Absent such hindsight reasoning, one of ordinary skill in the art

would not have been motivated to combine the references in the manner suggested by the Examiner.

Therefore, Appellants respectfully submit that the imposed rejection of claims 1 and 14 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess and Gamma is improper. Accordingly, reversal of the rejection is respectfully requested.

b) Claim 15

Claim 15 recites a system for managing computer system resources comprising means for generating a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message and means for generating a second command control vector associated with a second input message, the second command control vector identifying the same method object identified by the first command control vector, the method object containing one or more instructions for processing the second input message. The Office Action states that Burgess discloses these features and points to col. 4, lines 3-19 and col. 8, line 46 to col. 11, line 12 for support (Office Action – page 4). Appellants respectfully disagree.

Burgess at col. 4, lines 3-19, as discussed above with respect to claim 1, discloses invoking a target member function to process changes, such as changes in the value for a scroll bar when a slider on the scroll bar is moved. Invoking a target member function to process a change is not equivalent to generating first and second command control vectors associated with first and second input messages, where the second command control vector identifies the same method object identified by the first command control vector, as recited in claim 15.

Burgess at col. 8, line 46 to col. 11, line 12, as discussed above, refers to Code Table 3 that contains pseudocode for a sample scrollbar object. This portion of Burgess does not disclose or suggest means for generating a first command control vector and means for generating a second command control vector associated with first and second input messages, respectively, where the second command control vector identifies the same method object identified by the first command control vector and that the method object contains one or more instructions for processing the second input message, as recited in claim 15.

In particular, the Office Action indicates that the event class CIntEvent is equivalent to the claimed first message and the class CStringEvent is equivalent to the claimed second message (Office Action – pages 3-4). Appellants disagree with this analysis of Burgess.

The event class CIntEvent in Burgess is used to pass information stored as an integer to a target member function (Burgess – col. 8, lines 8-9). Burgess does not disclose or suggest means for generating a first command control vector associated with the event class CIntEvent, as would be required based on the Examiner's interpretation of CIntEvent as being equivalent to the claimed first input message. In contrast, the event class CIntEvent is merely used to pass information stored as an integer to a target member function.

Further, the class string CStringEvent is an event class used to pass messages with string data (Burgess – col. 8, lines 18-19). Burgess does not disclose or suggest generating a command control vector associated with the event class CStringEvent, as would be required based on the Examiner's interpretation of CStringEvent as being equivalent to the claimed second input message. In contrast, the event class CStringEvent is merely used to pass messages containing string data.

The Office Action admits that Burgess does not disclose means for providing a single copy of the method object for the first and second command vectors, but states that Gamma discloses creating a class to provide a single copy of the class object (Office Action – page 4). Gamma may disclose that a class in an object creation environment may have only one instance. Gamma, however, does not make up for the deficiencies in Burgess with respect to claim 15 discussed above. That is, Gamma does not disclose means for generating first and second command control vectors associated with first and second input messages, where the first and second command control vectors identify the same method object, as recited in claim 15.

Therefore, as a factual matter, the combination of Burgess and Gamma does not disclose or suggest each of the features of claim 15.

In addition, even if, for the sake of argument, the combination of Burgess and Gamma could reasonably be construed to disclose each of the features of claim 15, the Office Action does not provide the motivation required under 35 U.S.C. § 103 as to why it would have been obvious to combine these references.

For example, the Office Action further states that it would have been obvious to combine the teachings of Burgess and Gamma for the same reasons discussed above with respect to claim 1 (i.e., "one of ordinary skill in the art would have been motivated to use the mechanism of Gamma to achieve the hiding in Burgess") (Office Action – page 4). Appellants respectfully disagree.

As discussed above with respect to claim 1, the allegation that Burgess "desires to hide different implementations/functional prototypes of the target object from the source object" is not supported by Burgess at col. 4, lines 31-37. In addition, even if Burgess did disclose some desire

to hide different implementations/functional prototypes, as alleged, the Office Action has still not provided any objective motivation as to why it would have been obvious to combine Gamma with Burgess. The Office Action merely indicates that Gamma provides a mechanism for calling a target object with the same prototype and therefore, one of ordinary skill in the art would have been motivated to use the mechanism of Gamma to achieve the hiding in Burgess. This is merely a conclusory statement regarding an alleged benefit of the combination. Once again, the mere fact that one reference allegedly provides some missing disclosure with respect to a claim does not satisfy the requirements of 35 U.S.C. § 103 as to why it would have been obvious to combine these references.

Therefore, Appellants respectfully submit that the imposed rejection of claim 15 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess and Gamma is improper. Accordingly, reversal of the rejection is respectfully requested.

c) Claims 2 and 16

The arguments below use claim 2 as representative of the group of claims including claims 2 and 16. Claim 2 recites identifying, in the first command control vector, a communication link from which the first input message is received and identifying, in the first command control vector, a destination device for which the first input message is intended. The Office Action states that Burgess discloses these features and points to col. 4, lines 50-67 for support (Office Action – page 6). Appellants respectfully disagree.

Burgess at col. 4, lines 40-67 discloses that the class CObject tracks connections and controls the sending of messages. The class CObject may include a connection array that

contains connections for each output port of a run time object. Each entry of the connection array contains an identifier of the output port of the source object, an identifier of the target object and an identifier of the input port of the target object. Identifying an input port of the target object and an output port of the source object, as disclosed by Burgess, is not equivalent to identifying, in a first command control vector, a communication link from which the first input message is received and a destination device for which the first input message is intended. In contrast, this portion of Burgess merely discloses that entries in a connection array identify input ports of a target object and an output port of a source object.

Therefore, Appellants respectfully submit that the imposed rejection of claims 2 and 16 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess and Gamma is improper. Accordingly, reversal of the rejection is respectfully requested.

d) Claims 3 and 17

The arguments below use claim 3 as representative of the group of claims including claims 3 and 17. Claim 3 recites similar features as claim 2 with respect to the second input message. That is, claim 3 recites identifying, in the second command control vector, a communication link from which the second input message is received and identifying, in the second command control vector, a destination device for which the second input message is intended. The Office Action states that Burgess discloses these features and points to col. 4, lines 60-67 for support (Office Action – page 6). Similar to the discussion with respect to claim 2, identifying an output port of the source object and an input port of the target object in a connection array is not equivalent to identifying, in a second command control vector, a

communication link from which a message is received or a destination device to which the message is intended, as recited in claim 3.

Therefore, Appellants respectfully submit that the imposed rejection of claims 3 and 17 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess and Gamma is improper. Accordingly, reversal of the rejection is respectfully requested.

2. Claims 4 and 18 are patentable under 35 U.S.C. § 103 over Burgess in view of Gamma and further in view of the admitted prior art.

a) Claims 4 and 18

The arguments below use claim 4 as representative of the group of claims including claims 4 and 18. Claim 4 recites that the processing the first and second input messages comprises using a single copy of a script to process the first and second input messages. The Office Action states that the admitted prior art discloses a method object invoking a script and points to page 2, line 29 of Appellants' specification for support (Office Action – page 7).

Appellants' specification does disclose that a method object may be, for example, a script. This disclosure, however, does not read on the feature recited in claim 4. That is, the admitted prior art does not disclose that first and second current instructions identified in first and second command control vectors, respectively, are the same instruction or that this same instruction invokes a script.

The Office Action further states that the combined teachings of Burgess, Gamma and the admitted prior art would have provided a single copy of the script (Office Action – page 7). Appellants respectfully disagree.

None of the references, taken alone, or in combination discloses or suggests the use of first and second instructions that both invoke a single copy of a script to process first and second input messages, as required by claim 4. Therefore, even if the admitted prior art was combined with Burgess and Gamma, the claimed invention would not result.

Lastly, the Office Action states that the admitted prior art provides an interrupt mechanism (timer interrupt) to provide multi-tasking scheduling and one of ordinary skill in the art would have been motivated to use the interrupt mechanism of the admitted prior art to achieve multi-tasking in Burgess (Office Action – page 7). Appellants respectfully disagree.

First, Appellants' specification does mention that processing in multi-tasking systems may be interrupted and re-started (specification – page 3, lines 3-8). Claim 4, however, particularly recites that the first and second current instructions are the same instruction and the processing the first and second input messages comprises using a single copy of a script to process the first and second input messages. Therefore, even if the admitted prior art disclosed an interrupt mechanism, the admitted prior art does not disclose or suggest the features recited in claim 4.

Therefore, as a factual matter, even if the admitted prior art was combined with the combination of Burgess and Gamma, the claimed invention would not result.

Further, even if the combination of references was construed to disclose each of the claimed features, Appellants respectfully submit that the Office Action lacks the requisite motivation as to why one of ordinary skill in the art would have been motivated to combine these references.

For example, the Office Action states the admitted prior art provides an interrupt

mechanism to provide multi-tasking scheduling. Therefore, one of ordinary skill in the art would have been motivated to use the interrupt mechanism of APA to achieve multi-tasking in Burgess (Office Action – page 7).

Appellants note that no portion of any of the three references is pointed to as providing objective motivation for combining these three references. Secondly, the alleged motivation (i.e., to achieve multi-tasking in Burgess) is merely a conclusory statement regarding an alleged benefit of the combination. Such motivation does not satisfy the requirements of 35 U.S.C. § 103. In this respect, Appellants rely upon In re Deuel, 51 F.3d 1552, 34 USPQ2d 1210 (Fed. Cir. 1995), wherein it was held that generalizations do not establish the realistic motivation to modify a specific reference in a specific manner to arrive at a specifically claimed invention.

Therefore, Appellants respectfully submit that the imposed rejection of claims 4 and 18 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess, Gamma and the admitted prior art is improper. Accordingly, reversal of the rejection is respectfully requested.

3. Claims 5 and 6 are patentable under 35 U.S.C. § 103 over Burgess in view of Gamma and the admitted prior art and further in view of Carr.

a) Claim 5

Claim 5 recites that using a single copy of the script comprises identifying current script instructions in the first and second command control vectors for processing the first and second input messages, respectively; storing, in a first data object, data that is generated during execution of the script for the first command control vector; and storing, in a second data object,

data that is generated during execution of the script for the second command control vector.

The Office Action admits that Burgess does not disclose storing, in a first data object, data that is generated during execution of the script for the first command control vector; and storing, in a second data object, data that is generated during execution of the script for the second command control vector, as recited in claim 5. The Office Action, however, states that Carr discloses using a data object (value object) to store data generated during execution (return values) of a C++ program and points to pages 499-500 of Carr for support (Office Action – page 7). Even if Carr discloses such features, such a disclosure is not equivalent to storing, in a first data object, data that is generated during execution of the script for the first command control vector or storing, in a second data object, data that is generated during execution of the script for the second command control vector, as recited in claim 5.

Therefore, even if Carr was combined with the combination of Burgess, Gamma and the admitted prior art, the combination would not disclose each of the features of claim 5.

In addition, even if, for the sake of argument, the combination of Burgess, Gamma, the admitted prior art and Carr could reasonably be construed to disclose each of the features of claim 5, the motivation to combine Carr with the other disclosures does not satisfy the requirements of 35 U.S.C. § 103.

For example, the Office Action states that it would have been obvious to combine the teachings of Burgess as modified by Gamma and Carr because the former implements the teaching in language C++ and the latter details one version of the language C++. The Office Action further states that one of ordinary skill in the art would have been motivated to combine the teaching of Carr with the other references "since this would require less effort in integration"

(Office Action – pages 7-8). This statement is merely a conclusory statement and no portion of any of the references is pointed to as providing objective motivation for combining these disclosures. The mere fact that these references involve the C++ programming language does not provide the required motivation under § 103 as to why one of ordinary skill in the art would combine these references. Appellants again stress that generalizations, such as requiring less effort in integration, do not establish the requisite motivation under 35 U.S.C. § 103 to modify a reference in a specific manner. In re Deuel, supra.

Therefore, Appellants respectfully submit that the imposed rejection of claim 5 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess, Gamma, the admitted prior art and Carr is improper. Accordingly, reversal of the rejection is respectfully requested.

b) Claim 6

Claim 6 recites that the processing the first and second input messages comprises processing a number *n* of logical units of instructions for the first command control vector; interrupting processing of the first command control vector; and processing a number *m* of logical units of instructions for the second command control vector.

As to claim 6, Appellants note that this claim was rejected based on the combination of Burgess, Gamma, the admitted prior art and Carr. Carr, however, was not relied upon in the grounds of rejection. Appellants requested clarification as to the grounds of rejection in the previous responses. These requests were not addressed in the prior Office Actions or the current Office Action.

In any event, the Office Action states that the admitted prior art discloses processing n logic units of instructions of a first type, interrupting such processing and processing m logical units of instructions for a second type and points to page 3, lines 3-8 of Appellants' specification for support (Office Action – page 8). Appellants respectfully disagree.

Appellants' specification at page 3, lines 3-8 refers to multi-tasking systems which employ preemptive time-slice processing. Tasks in such systems are automatically interrupted after a predetermined period of time or after a specific instruction type. Processing may resume at a later point, preferably from where it was interrupted. This disclosure in Appellants' specification at page 3 is not equivalent to processing a number *n* of logical units of instructions for the first command control vector; interrupting processing of the first command control vector; and processing a number *m* of logical units of instructions for the second command control vector, as recited in claim 6.

Therefore, even if the admitted prior art was combined with Burgess, Gamma and Carr, the claimed invention would not result.

Further, even if, for the sake of argument, the combination of Burgess, Gamma, the admitted prior art and Carr could reasonably be construed to disclose each of the features of claim 6, the motivation to combine Carr with the other disclosures does not satisfy the requirements of 35 U.S.C. § 103.

For example, the Office Action states that it would have been obvious to alternate the processing of the first message/first command control vector and the processing of the second message/second command control vector and points to the discussion with respect to claim 4 for the motivation to combine the teachings (Office Action – page 8).

Appellants note that claim 4 did not recite features similar to those of claim 6. Therefore, the discussion with respect to combining features allegedly included in the admitted prior art with respect to claim 4 with Burgess and Gamma are not relevant with respect to the features recited in claim 6. Further, as discussed above with respect to claim 4, generalizations, such as those used in the motivation with respect to claim 4, do not establish the realistic motivation to modify a specific reference in a specific manner to arrive at a specifically claimed invention. In re Deuel, supra.

Therefore, Appellants respectfully submit that the imposed rejection of claim 6 under 35 U.S.C. § 103 for obviousness based on the combination of Burgess, Gamma, the admitted prior art and Carr is improper. Accordingly, reversal of the rejection is respectfully requested.

## IX. CONCLUSION

In view of the foregoing arguments, Appellants respectfully solicit the Honorable Board to reverse the Examiner's rejections of claims 1-6 and 14-18 under 35 U.S.C. § 103.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 13-2491 and please credit any excess fees to such deposit account.

Respectfully submitted,

By: _____
~~For~~ Glenn Snyder
Reg. No. 41,428

MICHAEL A. WRENN
REG. NO. 42,237

Date: 2/23/2004

WorldCom, Inc.
1133 19th Street NW
Telephone: 202.736.6522
Facsimile: 202.736.6382

APPENDIX

1. A method for managing computer system resources, comprising:

generating a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message, wherein the generating a first command control vector comprises:

identifying the method object in the first command control vector, and

identifying, in the first command control vector, a first current instruction of the method object, wherein the first current instruction is used to process the first input message;

generating a second command control vector associated with a second input message, the second command control vector identifying the same method object identified by the first command control vector, the method object containing one or more instructions for processing the second input message, wherein the generating a second command control vector comprises:

identifying the same method object in the second command control vector; and

identifying, in the second command control vector, a second current instruction of the method object, wherein the second current instruction is used to process the second input message;

providing a single copy of the method object for the first and second command control vectors; and

processing the first and second input messages using the single copy of the method object.

2. The method according to claim 1, wherein the generating a first command control

vector further comprises:

identifying, in the first command control vector, a communication link from which the first input message is received; and

identifying, in the first command control vector, a destination device for which the first input message is intended.

3. The method according to claim 2, wherein the generating a second command control vector further comprises:

identifying, in the second command control vector, a communication link from which the second input message is received; and

identifying, in the second command control vector, a destination device for which the second input message is intended.

4. The method of claim 3, wherein the first and second current instructions are the same instruction and the same instruction invokes a script, wherein the processing the first and second input messages comprises:

using a single copy of a script to process the first and second input messages.

5. The method of claim 4, wherein the using a single copy of the script comprises:

identifying current script instructions in the first and second command control vectors for processing the first and second input messages, respectively;

storing, in a first data object, data that is generated during execution of the script for the

first command control vector; and

storing, in a second data object, data that is generated during execution of the script for the second command control vector.

6. The method according to claim 1, wherein the processing the first and second input messages comprises:

processing a number *n* of logical units of instructions for the first command control vector;

interrupting processing of the first command control vector; and

processing a number *m* of logical units of instructions for the second command control vector.

14. A computer program product for permitting a computer system to manage computer system resources, said computer program product comprising:

a computer usable medium having computer readable program code means embodied in said medium for causing an application program to execute on the computer system, said computer readable program code means comprising:

a computer readable first program code means for causing the computer system to generate a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message, the first program code means further causing the computer system to identify a first current instruction of the method object and use the first current instruction to process the

first input message;

a computer readable second program code means for causing the computer system to generate a second command control vector associated with a second input message, the second command control vector identifying the same method object identified by the first command control vector, the method object containing one or more instructions for processing the second input message, the second program code means further causing the computer system to identify a second current instruction of the method object and use the second current instruction to process the second input message; and

a computer readable third program code means for causing the computer system to process the first and second input messages using a single copy of the method object.

15. A system for managing computer system resources, comprising:

means for generating a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message;

means for generating a second command control vector associated with a second input message, the second command control vector identifying the same method object identified by the first command control vector, the method object containing one or more instructions for processing the second input message;

means for providing a single copy of the method object for the first and second command control vectors; and

means for processing the first and second input messages using the single copy of the

method object.

16. The system according to claim 15, wherein the means for generating a first command control vector for a first input message further comprises:

means for identifying, in the first command control vector, a communication link from which the first input message is received;

means for identifying, in the first command control vector, a destination device for which the first input message is intended;

means for identifying the method object in the first command control vector; and

means for identifying, in the first command control vector, a first current instruction of the method object, wherein the first current instruction is used to process the first input message.

17. The system according to claim 15, wherein the means for generating a second command control vector associated with a second input message further comprises:

means for identifying, in the second command control vector, a communication link from which the second input message is received;

means for identifying, in the second command control vector, a destination device for which the second input message is intended;

means for identifying the same method object in the second command control vector; and

means for identifying, in the second command control vector, a second current instruction of the method object, wherein the second current instruction is used to process the second input message.

18. The system of claim 17, wherein the first and second current instructions are the same instruction and the same instruction invokes a script, wherein the means for processing the first and second input messages using the single copy of the method object further comprises:

means for using a single copy of a script to process the first and second input messages.